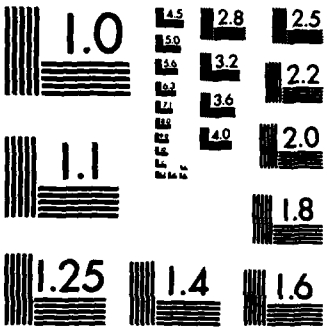
AD-A135 074 SUMMARY OF RESEARCH 15 JUNE 1982 TO 14 JUNE 1983 GRANT 1/1
AFOSR-81-0197(U) STATE UNIV OF NEW YORK AT STONY BROOK
DEPT OF COMPUTER SCIENCE A J BERNSTEIN JUL 83
UNCLASSIFIED AFOSR-TR-83-0930 AFOSR-81-0197 F/G 9/2 NL



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(2)

AD-A135 074

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83-0930	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SUMMARY OF RESEARCH, 15 JUNE 1982 TO 14 JUNE 1983, GRANT AFOSR-81-0197		5. TYPE OF REPORT & PERIOD COVERED INTERIM, 15 Jun 82-14 Jun 83 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Arthur J. Bernstein		8. CONTRACT OR GRANT NUMBER(s) AFOSR-81-0197
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Computer Science State University of New York at Stony Brook Long Island NY 11794		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE61102F; 2304/A2
11. CONTROLLING OFFICE NAME AND ADDRESS Mathematical & Information Sciences Directorate Air Force Office of Scientific Research /NM Bolling AFB DC 20332		12. REPORT DATE JUL 83 13. NUMBER OF PAGES 6
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The research performed under this grant centers on the concept of a network computer. By this the authors mean a network of computers (no shared memory) which can be programmed as if it were a single virtual machine using a high level distributed language. Work during this past year can be divided into three areas: (1) Distributed Algorithms; (2) Distributed Languages; and (3) An Implementation of Multicasting on a Network Computer. This report summarizes progress achieved during the past year.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AFOSR-TR- 83-0930

Summary of Research - 6/15/82 to 6/14/83
AFOSR81 0197
Professor Arthur J. Bernstein

The research performed under this grant centers on the concept of a network computer. By this we mean a network of computers (no shared memory) which can be programmed as if it were a single virtual machine using a high level distributed language. Work during this past year can be divided into three areas.

A. Distributed Algorithms

A number of distributed algorithms have been described in the literature for network computers. We are particularly interested in low level algorithms which, for example, might be used in a distributed operating system to support resource allocation or enhance reliability. The Arpanet routing algorithm [MW77] is an early example. More recently attention has been given to problems related to the implementation of distributed databases. Distributed algorithms for guaranteeing the consistency of such databases under concurrent operation and in the face of component failures have been developed (e.g., [Gr78], [BG81], [Th78], [Bo82], [BL82]). Other classes of algorithms deal with distributed deadlock detection (e.g. [MM79], [CM82]), load balancing in a distributed system (e.g. [BF81], [HW80], [Sh83]) and resource allocation (e.g. [ADD82], [Sm79]). Finally, algorithms related to techniques for organizing a distributed system such as the election of a leader [Ga82] and the enforcement of distributed synchronization [Sc82] fall in this category.

Algorithms of this type may involve one or more of the following features: replication of information, redundant computation, resiliency in the face of inconsistent information, communication failures or node failures. These algorithms are generally characterized by a rather high level of message type communication between the distributed processes. Processes generally do not wait for a response immediately after sending a message and in many cases there is no response at all. Furthermore, communication is frequently of a multicast or broadcast nature. In [CM82] each node relays a probe which it has received to a dynamically determined subset of other nodes. It does not reply to the sender. In [Ga82] a potential coordinator multicasts a message to all higher priority nodes and then awaits responses. In an implementation of two phase commit, the commit coordinator multicasts messages to all cohorts in a similar fashion.

Another aspect of communication in distributed algorithms is that a message need not always be addressed to a unique process; any one of a set of processes may be eligible to receive it. Thus for reasons of reliability, load sharing or to reduce the lengths of communication paths, duplicate servers may be distributed throughout a network. In the Pup internet [Bo82] name servers are duplicated at each gateway since there is at least one gateway on each net and it is up most of the time. In general, clients do not care which member of a set of identical

servers responds to a request.

Our major interest in this area has been to study the properties of a language suitable for programming this class of algorithms. This work will be described in the next section. As an outgrowth of the study, however, we have been examining some specific algorithms. In particular we have developed a distributed stable storage algorithm in which copies of a replicated database are distributed over nodes connected to a broadcast medium. This is a generalization of the work by Lampson [La81] on stable storage. In that model data is duplicated on independent storage devices at a single node and failures are categorized into those which are expected and those which are unexpected. The latter are disasters for which stable storage offers no protection. Algorithms are provided for failures in the former category which guarantee that data is preserved. Examples of expected failures are processor crashes (i.e. the processor is reset to some standard state), transient I/O errors, and a limited, spontaneous decay of information on mass storage. An example of an unexpected failure is the malicious behavior of a processor. There are two disadvantages of this approach. Although data may be preserved at a node where a processor crash has occurred, it will not be available during the time that the processor is down. Secondly, malfunctions of the processor other than simple crashes can destroy the data.

The algorithm we have developed is designed specifically for data reliability in a broadcast environment. It is loosely coupled in the sense that it is designed to function despite the fact that not all copies of the data need agree and not all processors may be functioning correctly. A significant aspect of the proposal is that if no errors occur the redundancy is largely transparent to the procedures for accessing the data, requiring no extra communication. Additional messages are required when errors are detected in copies of stored information. The system can handle a much wider class of errors than that described in [La81]. A significant part of the work is the development of a Markov model to describe the failure behavior of the system. The model relates various parameters of the algorithm to the mean time to data loss. A paper describing this work has been completed [Be83] and submitted for publication.

Other distributed algorithms which are being studied are the solution to the Byzantine Generals Problem [Do82] and protocols for updating multiple copy data bases where serial consistency is not required [FM82]. In the former case we are examining the effects of communication failure as opposed to processor failure and have formulated a weaker requirement for agreement. In the latter case we have developed a protocol for data base update with reduced communication requirements. This research is being carried out by Mr. Gene Wuu, a PhD student. It is still at an early stage.

B. Distributed Languages

This work is a continuation of the work performed in the previous year to develop a distributed language for the category of algorithms described in the previous section. That work culminated in the presentation of two papers during

this past year, one of which outlined the communication aspects of a distributed language [GB82], and the other a new switching technique for transporting messages in a local area network [AGBB83]. Two students working in this area completed their degree requirements during this period: Dr. David Gelernter received a PhD and is currently teaching in the Computer Science Department at Yale and Mr. Mauricio Arango received an MS and is currently working in industry.

A number of distributed languages have been described in the literature for implementing distributed algorithms. None, however, are oriented towards the type of applications described above. Some are transaction oriented and rely on remote procedure call as a means of communication [LS81], [Br78] (Ada [DOD80] also falls into this category). Others are more flexible in that asynchronous message passing is provided as well [SY83], [Co79], [An81]. Others are primarily message oriented [Li79], [Ho78], [Fe79]. None, however, support multicast communication or allow a generalized addressing scheme in which any one of a set of processes - which might be distributed through the net - can be the recipient of a message.

Our work differs from that of other proposals in this area in that instead of addressing a message to a particular process, a message is addressed to a name which is visible in some region of a distributed program containing both the sender and the intended receiver(s). If the message is sent in unicast mode then any process within the region is eliglble to receive it; if it is sent in multicast mode then multiple processes in the region may copy the message. Thus name based addressing naturally integrates both concepts. Messages may be deleted when they are no longer relevant. For example, in the contract net protocol [Sm79] outstanding messages requesting bids should be deleted when the bid period is over.

This project is now continuing under the direction of a new PhD student, Mr. Mustaque Ahamad. Our initial work consisted of a refinement of Dr. Gelernter's proposal based on an examination of distributed algorithms taken from the literature. In particular, communication statements dealing with multicast have been modified, language structures for dynamically establishing communication paths have been developed and an implementation schema suitable for a general communication environment has been designed. A syntax has been developed which includes exception handling. A technical report on this subject will be released shortly.

One aspect of this work is the development of formal semantics for the communication constructs of the language. We are preparing to extend the work described in [SS82] to deal with name based addressing, multicast and message withdrawal. Initial investigations indicate that this will bear some relationship to the work on temporal logic which was completed this past year under grant support. Dr. Paul Harter completed the requirements for a PhD in December, 1982 with a thesis in this area and is currently teaching in the Computer Science Department at the University of Colorado.

C. An Implementation of Multicasting on a Network Computer

A project has been in progress for the past year to design (and ultimately implement) multicast communication on a network computer. This work is being done in collaboration with Prof. Larry Wittie and one of his students, Mr. Ariel Frank. The target system consists of some Motorola 68000 computers (a few of which are actually SUN workstations) connected by ethernet pathways. Some grant equipment money was used for this during the past year. Additional items were purchased with funds from an NSF equipment grant. That award was partially based on results achieved with AFOSR support.

The problem is to design a communications kernel for each node in the network which will support multicast addressing. This is a generalization of the directed broadcast scheme of [Bo82]. The goal is to assure that a multicast message is physically broadcast on a subset of ethernets which covers all nodes containing potential receivers. Implementation will be based on the multicast addressing provided in ethernet controllers. Each controller responds to a set of logical addresses whose membership may be dynamically changed. This work meshes closely with the language project described in the previous section. Names will be mapped into logical addresses. The contents of the set at a particular node will correspond to the set of names being used by modules allocated to that node. Implementation of the multicasting structure is being done in Modula-2. Successful completion of this work will yield an environment which will support a distributed language realized as an extension to Modula-2.

No patents have been requested on this research.

References

[AGBB83] M. Arango, D. Gelernter, H. Badr and A. Bernstein, "Staged Circuit Switching for Network Computers", to appear in Proc. of ACM SIGCOMM 83 Symp.: Communications, Architectures and Protocols, Austin, Texas, Mar 1983.

[ADD82] G. Andrews, D. Dobkin and P. Downey, "Distributed Allocation with Pools of Servers", Proc. of ACM SIGACT-SIGOPS Symp on Principles of Distributed Computing, Ottawa, Canada, Aug 1982.

[An81] G. Andrews, "Synchronizing Resources", ACM Trans on Programming Languages and Systems, vol 3, Oct 1981.

[BF81] R. Bryant and R. Finkel, "A Stable Distributed Scheduling Algorithm", Proc 2nd Int'l Conf on Distributed Computing, 1981.

[Be83] A. Bernstein, "A Loosely Coupled Distributed Algorithm for Reliably Storing Data", Tech. Report #83/049, Dep't. of Computer Science, State Univ. of

New York, Stony Brook, NY, Jul 1983

[BG81] P. Bernstein and N. Goodman, "Concurrency Control in Distributed Database Systems", ACM Computing Surveys, vol 13, Jun 1981.

[BL82] H. Breitwieser and M. Leszak, "A Distributed Transaction Processing Protocol Based on Majority Consensus", Proc. of ACM SIGACT- SIGOPS Symp on Principles of Distributed Computing, Ottawa, Canada, Aug 1982.

[Bo82] D. Boggs, "Internet Broadcasting", PhD. Thesis, Dept. of Elect. Engin., Stanford Univ, Stanford CA., Jan 1982.

[Br78] P. Brinch Hansen, "Distributed Processes: A Concurrent Programming Concept", Comm ACM, vol 21, Nov 1978.

[CM82] K. Chandy and J. Misra, "A Distributed Algorithm for Detecting Resource Deadlocks in Distributed Systems", Proc. of ACM SIGACT-SIGOPS Symp on Principles of Distributed Computing, Ottawa, Canada, Aug 1982.

[Co79] R. Cook, "*MOD: A Language for Distributed Programming", Proc. First Int'l Conf on Distributed Computing Systems, Oct 1979.

[Do82] D. Dolev, "The Byzantine Generals Strike Again", J. of Algorithms, vol 3, 1982.

[DOD80] Reference Manual for the Ada Programming Language, Dept of Defense, Jul 1980.

[Fe79] J. Feldman, "High Level Programming for Distributed Computing", Comm ACM, vol 22, Jun 1979.

[FM82] M. Fischer and A. Michael, "Sacrificing Serializability to Attain High Availability of Data in an Unreliable Network", Proc. ACM Symp. on Principles of Database Systems, Los Angeles, CA., Mar 1982.

[Ga82] H. Garcia-Molina, "Elections in a Distributed Computing System", IEEE Trans on Computers, vol C-31, Jan 1982.

[GB82] D. Gelernter and A. Bernstein, "Distributed Communication via Global Buffer", Proc. of ACM SIGACT-SIGOPS Symp on Principles of Distributed Computing, Ottawa, Canada, Aug 1982.

[Gr78] J. Gray, "Notes on Database Operating Systems", in Lecture Notes in Computer Science, Operating Systems: An Advanced Course, vol 60, Springer-Verlag, 1978.

[Ho78] C. Hoare, "Communicating Sequential Processes", Comm ACM, vol 21, Aug 1978.

[HW80] R. Halstead and S. Ward, "The MUNET: A Scalable Decentralized Architecture for Parallel Computation", Proc 7th Annual Symp on Computer Architecture, 1980.

[La81] B. Lampson, "Atomic Transactions", in Lecture Notes in Computer Science, Distributed Systems - Architecture and Implementation, vol 105, Springer-Verlag, 1981.

[Li79] B. Liskov, "Primitives for Distributed Computing", Proc 7th Symp on Operating Systems Principles, Dec 1979.

[LS81] B. Liskov and R. Scheifler, "Guardians and Actions: Linguistic Support for Robust Distributed Programs", Computation Structures Group Memo 210, MIT, 1981.

[MM79] D. Menasce and R. Muntz, "Locking and Deadlock Detection in Distributed Databases", IEEE Trans on Software Engineering, vol SE-5, May 1979.

[MW77] J. McQuillan and D. Walden, "The ARPA Network Design Decisions", Computer Networks, vol 1, Aug 1977.

[Sh83] L. Sha, E. Jensen, R. Rashid and J. Northcutt, "Distributed Cooperating Processes and Transactions", Proc. ACM SIGCOMM'83, Univ of Texas, Austin, Texas, Mar 1983.

[SS82] R. Schlichting and F. Schneider, "Understanding and Using Asynchronous Message Passing", Proc. of ACM SIGACT-SIGOPS Symp on Principles of Distributed Computing, Ottawa, Canada, Aug 1982.

[Sc82] F. Schneider, "Synchronization in Distributed Programs", ACM Trans. on Programming Languages and Systems, vol 4, Apr 1982.

[Sm79] R. Smith, "The Contract Net Protocol", First Int'l. Conf. on Distributed Computing Systems, Huntsville, Ala., Oct 1979.

[SY83] R. Strom and S. Yemini, "NIL: An Integrated Language and System for Distributed Programming", Proc. SIGPLAN '83 Symp. on Programming Language Issues in Software Systems, San Francisco, CA, Jun 1983.

[Th78] R. Thomas, "A Solution to the Concurrency Control Problem for Multiple Copy Databases", IEEE Compcon'78, Apr 1978.

END

FILMED

12-83

DTIC